

## INTRUÇÕES

Usaremos o SGBD Neo4j.

### 1. Baixar o Neo4j:

Baixar o .tar.gz no site do próprio Neo4J, através do link: <https://goo.gl/nqmiB8>. Após baixar o arquivo você deve descompactá-lo. Sugestão de comando para descompactação:

```
tar -xf neo4j-community-3.4.0-unix.tar.gz
```

Obs: Caso encontre problemas ao baixar diretamente do site, acesse o seguinte link alternativo: <https://goo.gl/7fEom9>

### 2. Executar: No terminal digite: <local onde descompactou o neo4j>/bin/neo4j start

### 3. Acessar no navegador: localhost:7474

### 4. No primeiro acesso usa as credenciais default:

*usuário:* neo4j

*senha:* neo4j

### 5. Digitar nova senha

**Importante:** Cada aluno acessa sua própria instância do banco de dados.

As duas partes do lab contém exemplos adaptados do site neo4j.com. As consultas da parte 1 estão acompanhadas de respostas. Tente resolve-las sem olhar as respostas.

## ----- PARTE 1: CONSULTAS DE EXEMPLO

### 1. Crie você como um nó.

```
CREATE (you:Person {name:"You"})
RETURN you
```

### 2. Encontre você mesmo e expresse que você gosta do BD Neo4j.

```
MATCH (you:Person {name:"You"})
CREATE (you)-[like:LIKE]->(neo:Database {name:"Neo4j" })
RETURN you,like,neo
```

### 3. Crie 5 amigos para você. É possível inserir laços de repetição e estruturas de dados com os comandos Cypher.

```
MATCH (you:Person {name:"You"})
FOREACH (name in ["Johan","Rajesh","Anna","Julia","Andrew"] |
  CREATE (you)-[:FRIEND]->(:Person {name:name}))
```

### 4. Obtenha seus amigos.

```
MATCH (you {name:"You"})-[:FRIEND]->(yourFriends)
RETURN you, yourFriends
```

### 5. Crie um amigo de um amigo seu que trabalhe com Neo4j.

```
MATCH (neo:Database {name:"Neo4j"})
```

```
MATCH (anna:Person {name:"Anna"})
CREATE (anna)-[:FRIEND]->(:Person:Expert {name:"Amanda"})-
[:WORKED_WITH]->(neo)
```

**6. Encontre a pessoa mais próxima em sua rede de relacionamentos que possa te ensinar Neo4j (limite sua profundidade de busca a 3 níveis).**

```
MATCH (you {name:"You"})
MATCH (expert)-[:WORKED_WITH]->(db:Database {name:"Neo4j"})
MATCH path = shortestPath( (you)-[:FRIEND*..3]-(expert) )
RETURN db,expert,path
```

**7. Apague todos os dados que você criou.**

```
MATCH (n)
DETACH DELETE n
```

-----

## PARTE 2: CONSULTAS AVANÇADAS

Vamos agora trabalhar com um maior volume de dados.

Para criar e popular seu banco de dados, baixe o arquivo `database.txt`, disponível em: <https://goo.gl/MmhpMq>.

Execute o conteúdo do arquivo no terminal dentro do browser.

**1. Obter o diagrama de dados implícito desses dados. Identifique quais as entidades e quais relacionamentos.**

**2. Obter o nome de todos os pilotos**

```
MATCH (driver:Driver) RETURN driver.name
```

**3. Obter cada piloto e seus pontos.**

```
MATCH (driver:Driver)-[f:FINISHED]->(circuit:GrandPrix)
RETURN driver.name AS Name,SUM(f.points) AS Points
ORDER BY Points DESC
```

**4. Obter os pontos de cada construtor.**

```
MATCH (constructor:Constructor)<-[:BELONGS_TO]-(driver:Driver)-
[f:FINISHED]->(circuit:GrandPrix)
RETURN constructor.name as Constructor,SUM(f.points) as Points
ORDER BY SUM(f.points) DESC
```

**5. Obter o piloto que correu novamente por um construtor que previamente abandonou.**

```
MATCH (driver:Driver)-[:BELONGS_TO]->(constructor:Constructor)<-
[:BELONGED_TO]-(driver)
RETURN driver.name AS Name,constructor.name AS Constructor
```

**6. Obter os construtores agrupados por Fabricantes/Modelos que utilizam (Engines).**

```
MATCH (team:Constructor)-[:HAS]->(engineer:Engine)
RETURN engineer.name AS Engine, collect(team.name) AS Teams
ORDER BY Engine
```

**7. Obter por país a lista de pilotos, a média de pontos por corrida e o total de pontos.**

```
MATCH (country:Country)<-[:COUNTRY_ORIGIN]-(driver:Driver)-
```

```
[f:FINISHED]->(circuit:GrandPrix)
RETURN country.name as Country,collect(distinct(driver.name)) as
Drivers,round(AVG(f.points)) as AveragePerRace ,SUM(f.points)as
TotalPoints
ORDER BY AveragePerRace DESC, TotalPoints DESC
```

**8. Obter para cada piloto quantas vezes ele obteve uma determinada colocação.**

```
MATCH (driver:Driver)-[f:FINISHED]->()
WHERE f.position=f.position
RETURN driver.name AS Name, count(f.position) AS Times, f.position
AS
Position
ORDER BY driver.name, Times DESC
```

**9. Obter quantas vezes o piloto Fernando Alonso obteve cada colocação.**

```
MATCH (driver:Driver{name:"Fernando Alonso"})-[f:FINISHED]->()
WHERE f.position=f.position
RETURN driver.name AS Name, count(f.position) AS Times, f.position
AS
Position
ORDER BY driver.name, Times DESC
```

**10. Obter os pilotos que são do mesmo país que suas equipes.**

```
MATCH (driver:Driver)-[b:BELONGS_TO|BELONGED_TO]-
>(cons:Constructor)-
[co:COUNTRY_ORIGIN]->(country)<-[:COUNTRY_ORIGIN]-(driver)
RETURN driver.name AS Driver,cons.name AS Constructor,country.name
AS
Country
```